Comparing Approaches to Dravidian Language Identification

Jauhiainen, Tommi

2021

Jauhiainen , T , Ranasinghe , T & Zampieri , M 2021 , ' Comparing Approaches to Dravidian Language Identification ' , Workshop on NLP for similar languages, varieties and dialects , 20/04/2021 - 20/04/2021 .

http://hdl.handle.net/10138/330685

COMPARING APPROACHES TO DRAVIDIAN LANGUAGE IDENTIFICATION

Tommi Jauhiainen
University of Helsinki
Rochester Institute of Technology

Tharindu Ranasinghe
University of Wolverhampton

Marcos Zampieri
Rochester Institute of Technology

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
HUMANISTINEN TIEDEKUNTA
HUMANISTISKA FAKULTETEN
FACULTY OF ARTS

INTRODUCTION

This poster describes the submissions by team HWR to the Dravidian Language Identification (DLI) shared task organized at VarDial 2021 workshop. Discriminating between similar languages (e.g. Bulgarian and Macedonian or Croatian and Serbian), language varieties (e.g. Brazilian and European Portuguese), and dialects is one of the main challenges in automatic language identification (LI) in texts. We took this opportunity to evaluate the performance of two models for this task, a Naive Bayes (NB) classifier using adaptive language models and a transformers-based system.

DLI SHARED TASK

The data set provided by the DLI organizers contains a total of 22,164 YouTube comments written in a mix of English and one of the South Dravidian languages: Kannada, Malayalam, and Tamil. In addition to the target languages, the data included comments in other languages as well. It was divided into 16,674 instances for training and 4,590 instances for testing.

In order to evaluate and compare our methods using the training data, we divided the training data into training and development portions. For training, we used the first 90% of comments for each language and the rest was set aside for development. This way, we retained the original distribution of different labels as the provided training data seemed not to be in a random order.

METHODS

Before adding the language adaptation feature, the NB classifier was evaluated with several combinations of character *n*-grams and penalty modifiers. The best macro F1 score on the development data, 0.8609 was attained using character *n*-grams from 2 to 6.

The adaptation method uses several parameters which have to be optimized using the training and the development material. The first parameter is the number of splits the whole material to be identified is divided in. The actual division into splits happens after each time the test set is preliminarily identified and ordered so that the mystery texts with the highest difference between the log probabilities of the most probable and the second most probable language are on the top of the list. When incorporating new information from the text to be identified, the highest split is processed first. After its information has been added to the language models, all the remaining mystery texts are again preliminarily identified and divided into same sized splits. Again the information from the best split is incorporated into language models and so on, until all the splits have been processed. Using the adaptation method, the F1 score improved only slightly to 0.8663.

The system for our second submission was based on pretrained transformer models: multilingual BERT and XLM-RoBERTa (XLM-R). We pass the sentence through the transformer model and add a softmax layer on top of the [CLS] token as a normal classification architecture with transformers. We fine-tune all the parameters from transformer model as well as the softmax layer jointly by maximising the log-probability of the correct label. This architecture has been used widely in many text classification tasks that includes Malayalam code-mix texts too. We did not perform any pre-processing to this architecture. Considering the pretrained transformer models that supports Kannada, Malayalam and Tamil. In the initial experiments it achieved a macro F1 score of 0.785, which was considerably lower than the 0.861 gained by the basic NB model even though it used pretrained models as opposed to the NB which was using only the data provided for the DLI task.

RESULTS AND CONCLUSIONS

Table shows the results of the shared task. Our first run was clearly better than the second and not far behind the results of the LAST-team.

Rank	Team	Run	Macro F_1
1	LAST	1	0.93
	LAST	2 & 3	0.92
2	**HWR**	**1**	**0.92**
2	NYAEL	1	0.92
	NAYEL	2	0.91
4	Phlyers	1	0.89
	Phlyers	2	0.89
	HWR	**2**	**0.89**
	NAYEL	3	0.84

Even though the difference in performance between the NB model and the transformers was only 3 percentage points in the test set, the fact that the transformers did not outperform the NB classifier deserves special attention. One of the reasons to the inferior performance of the pretrained models is probably that the comments contained code-mixed sentences kind of which the pretrained language models like BERT and XLM-R had not seen before. However, our results are well in line with the general trend of deep learning methods not being overtly competitive in language identification tasks.

ACKNOWLEGMENTS

The research presented in this article has been partly supported by The Finnish Research Impact Foundation Tandem Industry Academia funding in cooperation with Lingsoft.

CONTACT INFORMATION

For the first author:
firstname.lastname@helsinki.fi